IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 6, 2018, accepted April 9, 2018, date of publication May 17, 2018, date of current version June 20, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2834912

ARAG: A Routing Algorithm Based on Incentive Mechanisms for DTN With Nodes' Selfishness

HEZHE WANG, HUIQIANG WANG, FANGFANG GUO, GUANGSHENG FENG, AND HONGWU LV

College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Corresponding author: Huiqiang Wang (wanghuiqiang@hrbeu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61502118, Grant 61402127, and Grant 61370212, in part by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2016ZX03001023-005, in part by the Natural Science Foundation of Heilongjiang Province in China under Grant F2016028, Grant F2016009, and Grant F2015029, and in part by the Ph.D. Student Research and Innovation Fund of the Fundamental Research Funds for the Central Universities under Grant HEUGIP201809.

ABSTRACT A delay tolerant network (DTN) is a mobile ad hoc network with extremely limited network resources. When DTN is applied in the urban environment, nodes are not willing to consume their limited resources to forward messages for other nodes, or they prefer a relay node that encounters the destination node with a higher probability to forward messages for themselves owing to the characteristics of sociality and selfishness of nodes, which results in poor network performance. To address this issue, a routing algorithm based on incentive mechanisms was proposed in this paper, which is called ARAG. In the process of forwarding message, the algorithm determines the probability of messages received by relay nodes based on the resource consumption of the sender node. Simultaneously, when a source node generates a message, the threshold of copies is set based on the consumption of its resources, and the total number of the same message in the network will not exceed the threshold. Simulation results show that the algorithm is superior to the existing Epidemic algorithm, Prophet algorithm, and GTMEA algorithm in terms of the delivery ratio, the average delay, and network overhead.

INDEX TERMS DTN, routing algorithm, buffer management, incentive mechanism.

I. INTRODUCTION

Delay tolerant network (DTN) [1] originally originated from interplanetary networks, mainly used to handle communication between satellites and base stations [2]. With the continuous development of network, DTN has been used for several applications, such as military networks [3], vehicle networks [4], [5], and wildlife tracking [6], [7]. Compared with the traditional network, DTN has intermittent connection, limited resources, lack of end-to-end connection path, and other characteristics [8]. In the process of forwarding message, if a message in the node cannot be forwarded to the destination node in time, the node does not immediately discard the message, rather it is stored in the node buffer until encounters the destination node of the message or other relay nodes that can forward the message [4].

A node does not only serve as a relay to carry or forward messages for other nodes during the process of forwarding message, but also attempts to forward messages of itself to other nodes. However, a node's buffer and energy are extremely limited in DTN [9], so that it prefers other relay nodes to forward messages for itself rather than self-consuming resources to forward messages for other nodes.

With the continuous development of low-cost mobile terminal equipment, the application of DTN in the urban environment has become a hot topic for DTN research in recent years [10]. In the urban environment, since nodes have sociality, the node's selfishness is more obvious, such as nodes are not willing to consume its own resources to forward messages for other nodes or they prefer a relay node that encounters the destination node with a higher probability to forward messages for themselves. Thus, the node's selfishness will significantly affect the performance of the network [11]. Therefore, it is very important to propose an efficient routing algorithm for DTN when it is applied in the urban environment.

In this paper, we propose a DTN routing algorithm based on incentive mechanisms to solve the selfishness of nodes in the urban environment. By encouraging a node to send messages to a relay node or carry messages for other nodes, the probability of messages in its buffer received by other relay nodes in the process of message forwarding can be increased. The threshold of copies of the message was simultaneously set to limit the number of copies of the message, thus ensuring efficient use of network resources.

The rest of this paper is organized as follows. Related works are presented in section 2, the network model is described in section 3, the ARAG algorithm is presented in section 4, while the simulation experiments and results are described in section 5. Finally, the conclusions are summarized in section 6.

II. RELATED WORK

Because DTN is characterized by intermittent connection, limited resources, and lack of end-to-end connection path, the traditional "store-forward" routing mode is not suitable for DTN. DTN relies on the movement of nodes to forward messages based on the routing mode of "store-carry-forward" [1], [12]. In the past few years, many research topics on DTN have been investigated; among them, the routing algorithm is a hot research topic. To cope with characterizes of the network, several routing algorithms have been proposed, such as epidemic routing [13] and probabilistic routing [14]. Most of these routing algorithms are multi-copy routing. In the process of forwarding or storing message, multi-copy routing will consume a large amount of network energy resources and buffer resources, resulting in fast depletion of network resources and subsequently, a decline in network performance [15]. Such decline in the performance is particularly obvious when DTN is applied in the urban environment owing to the selfishness of the node. Therefore, the design of a routing algorithm to solve the selfishness of nodes is an important objective when DTN is applied in the urban environment.

There are several existing routing algorithms for solving the selfishness of nodes. El-Azouzi *et al.* [16] presented a framework to achieve a tradeoff between successful data delivery probability and energy costs. They also investigated the impact of the proportion of the surface covered by both regions on the Nash equilibrium and the price of anarchy, and implemented a fully distributed algorithm that can be employed to achieve convergence to the Nash equilibrium. Srinivasan et al. [17] assumed that users will not always be willing to expend their energy resources to relay traffic generated by other users in wireless ad hoc networks. Thus, they applied the Nash game theory to address the problem of cooperation among energy constrained nodes in wireless ad hoc networks. Michiardi *et al.* [18] used a cooperative game approach and a non-cooperative game approach to evaluate the effectiveness of the CORE (collaborative reputation) mechanism. Saad *et al.* [19] proposed a game model to analyze cooperative decisions between multiple effective communities. Naserian *et al.* [20] proposed a routing algorithm based on forwarding game. Gholap *et al.* [21] presented a multi-receiver incentive-based dissemination (MuRIS) scheme that motivates node to forward the messages to the next node by giving it some reward. Brun *et al.* [22] presented an incentive mechanism for DTN, a reward is given only to the relay that is the first one to deliver the message to the destination. Seregina *et al.* [23] proposed a reward mechanism to incentive relays to sacrifice their memory and battery on DTN relaying operation. Chen *et al.* [11] proposed an incentive scheme called multicent for DTN routing that encourages nodes to cooperate and can realize different performance objectives and adjustable QoS for packets of specific sources, destinations, or source-destination pairs.

At the same time, in order to solve the problem of nodes' selfishness in the network, Mao and Zhu [24] proposed an energy-aware routing algorithm in which the node forwards more messages to receive more services from other nodes. The algorithm can effectively improve the performance of the network by considering the energy resources. However, the node's buffer resources were not considered. In addition, the algorithm did not consider how to control the amount of messages in the network.

Based on the above analysis, this paper proposes a DTN routing algorithm based on incentive mechanisms. The contribution of this article is as follows: (1) We define a contribution function to measure node's resource consumption, when the resource consumption value is greater than 0, it will increase the value of the contribution function, so that the messages in their buffer will have a greater probability of being received in process of forwarding. (2) Based on the contribution function value, we design a buffer management mechanism to solve the issue of extremely limited node resources can be used efficiently.

III. NETWORK MODEL

In this paper, we model the network as an undirected graph $G = (V, E)$, where V is a set of nodes, E is a set of edges, which represents connections between the nodes, and arbitrarily defined two nodes in the network, where a connection between two nodes means that communication is within each other's communication radius. For any node N in the network to maintain a list, $\mathrm{V}(N) = \{C_t(N), P_t(N), \Delta E_t(N), \Delta M_t(N), TC(N), p_N^M\}$. $C_t(N)$ is the contribution value of node N at time t, as given in Equation (1); $P_t(N)$ is the resource consumption value of node N at time t, as given in Equation (2); $\Delta E_t(N)$ is the energy consumption value of node N at time t, as given in Equation (3); $\Delta M_t(N)$ is the buffer consumption value of node N at time t, as given in Equation (4); and $TC(N)$ is a matrix that records the times node N forwards messages to relay nodes or destination nodes, as given in Equation (5); p_N^M represents the probability of node M encounters with node N, a detailed description of each node N records its probability of encountering with other nodes will be given in section 4.2.

$$C_t(N) = (1 - \alpha) \times C_{t-1}(N) + \alpha \times P_t(N), \tag{1}$$

where α is the adjustment factor. The term $P_t(N)$ is given as:

$$P_t(N) = \beta \times \Delta E_t(N) + (1 - \beta) \times \Delta M_t(N), \tag{2}$$

where β is the adjustment factor, $\Delta E_t(N)$ and $\Delta M_t(N)$ are described in Equation (3) and Equation (4), respectively.

$$\Delta E_t(N) = \sum\nolimits_0^t \varepsilon^N(E_c, t), \tag{3}$$

$$\Delta M_t(N) = \sum\nolimits_0^t \varepsilon^N(M_c, t), \tag{4}$$

where ε is the event statistics function, E_c is expressed as an energy consumption event, and M_c is expressed as a buffer usage event.

$$TC(N) = [TC_N^1, TC_N^2, \ldots, TC_N^m], \tag{5}$$

where TC_N^m is the number of times that node N forwards messages to the destination node of message m.

Each message m in the network maintains a list of attributes L(m), L(m) $= (S_m, D_m, C(S_m), N_m)$, where S_m is the source node of the message m, D_m is the destination node of the message m, $C(S_m)$ is the contribution value of the source node of the message m when message m is generated, and N_m is the number of copies when the message m is carried by any node. When N_m is greater than one, the message m only occupies one message buffer space of node.

IV. ARAG ALGORITHM

In this paper, we define the contribution function of node in section 3. The contribution value is used for determining the probability of messages received by relay nodes when forwarding messages. Based on the contribution value, the number of copies of the message is set when it is generated. A detailed description of routing and buffer process is given in the following section.

A. ROUTING POLICY

Step 1: It is assumed that node N has messages that require to forward. When node N and node M encounter each other, if $\Delta E_M - \Delta E_N \geq \delta_1$ and $\Delta M_M - \Delta M_N \geq \delta_2$, node N requests to establish a connection with node M. Otherwise, node N does not requests to establish a connection with the node M.

Where ΔE_N and ΔE_M are the remaining energies of nodes N and M, respectively, ΔM_N and ΔM_M denote the remaining buffer size of nodes N and M, respectively, and δ_1 and δ_2 are the adjustment factors.

Step 2: After node N and node M establish a connection, for any message m in the buffer of node N, calculate η_N^m and η_M^m according to equation (6). When $\eta_N^m < \eta_M^m$, node N forwards the message m to node M, node M receives the message of node N with a probability of $P(C_t(N))$. $P(C_t(N))$ is calculated as shown in equation (7). Otherwise, message m will not be forwarded.

$$\eta_N^m = \frac{TC_N^m}{TC_N^{total}}, \tag{6}$$

$$P(C_t(N)) = 1 - \frac{1}{e^{rc_t(N)}}, \tag{7}$$

where TC_N^m is the number of times that node N forwards messages to the destination node of m, TC_N^{total} is the total number of times that node N forwards messages, and γ is the adjustment factor.

B. BUFFER MANAGEMENT

In this paper, a buffer management mechanism is proposed to solve the issue of extremely limited network resources can be used efficiently. For the source node of message m, when the message is generated, the number of copies assigned to message m is N_m^S, that is to say, the maximum number of copies of message m is N_m^S in the network. When the number of copies of message m in a node is greater than one, the message m occupies only one message space of the node. The N_m^S is calculated as shown in equation (8):

$$N_m^S = \mathrm{K} \times C_t(S_m), \tag{8}$$

where K is the adjustment factor. When forwarding messages, the number of copies to be forwarded to the next relay node M is AL, as shown in equation (9):

$$AL = \left\lfloor N_m^N \times \frac{C_t(M)}{C_t(N) + C_t(M)} \times p_M^{dest} \right\rfloor, \tag{9}$$

where N_m^N is the number of copies of the message m contained in node N. When the number of copies of the message m in node N is equal to one, node N forwards the message m to node M and still not store the message m any more. p_M^{dest} is the probability of the node M encounters with the destination node of the message m.

Among them, each node N records its probability of encountering with other nodes accord to the following rules:

Such as p_N^M represents the probability of node M encounters with node N, and $M \in V(G) - \{N\}$. For each node N, the initial value of the encounter probability p_N^M is set to $1/(|V| - 1)$, where $|V|$ represents the number of nodes in the network. The update rule of p_N^M is as follows: When node N encounters node M again, p_N^M will add one. Then, the encounter probability for all nodes kept at N belonging to $V(G) - \{N\}$ is re-normalized [26].

When the node buffer overflows, the message with the least number of copies of message in the node buffer is discarded. If the number of copies of multiple messages is the least at the same time, discard the message generated by its source node which has the least contribution value.

C. INCENTIVE PROCESS IN ARAG

In this section, we will provide a detailed description of incentive process, where the node resource consumption matrix is listed in Table 1:

TABLE 1. Node resource consumption matrix.

(N, M)	R	UR
S	(S_3, S_1)	(S_4, -)
US	(-, S_2)	(- , -)

For node M as a recipient, M can choose to receive or not receive messages. When receiving messages, the resource consumed by node M depends on whether node N sends messages or not. When a message m is sent by node N, the resource consumed by node M is recorded as S_1. When no message is sent by node N, the resources consumed by node M is recorded as S_2. The specific description of S_1 and S_2 are shown in equations (10) and (11).

$$S_1 = P_t(M) = \beta_1 \times (E^{Lis} + Er^m) + (1 - \beta_1) \times Mr^m, \quad (10)$$
$$S_2 = P_t(M) = \beta_1 \times E^{Lis}, \quad (11)$$

where β_1 is the adjustment factor, E^{Lis} is the energy consumed by node M monitoring channel, Er^m is the energy consumed by node M receives the message m, and Mr^m is the size of the buffer change of node M after node M receives the message m.

For node N as a sender, N can choose to send messages or not to send messages. When sending messages, the resource consumed by node N depends on whether node M receives the message or not. When node M receives the message m which is sent by node N, the resource consumed by node N is denoted as S_3. When node M does not receive the message m sent by node N, the resource consumed by node N is denoted as S_4. The specific description of S_3 and S_4 are shown in equations (12) and (13).

$$S_3 = P_t(N) = \beta_2 \times (E^{Lis} + Es^m) + (1 - \beta_2) \times Ms^m, \quad (12)$$
$$S_4 = P_t(N) = \beta_2 \times E^{Lis}, \quad (13)$$

where β_2 is the adjustment factor, E^{Lis} is the energy consumed by node N monitoring channel, Es^m is the energy consumed by node N to send a message m, and Ms^m is the size of the buffer change of node N after node N sends a message m.

As can be seen from Table 1, when node N chooses to send a message and node M chooses not to receive the message, the resource consumption value of node M is zero; If node M chooses to receive the message, its resource consumption value is S_1, which is greater than zero.

When node N chooses not to send a message and node M chooses not to receive the message, the resource consumption value of node M is zero. If node M chooses to receive a message, its resource consumption value is S_2, which is greater than zero. Thus, node M should choose to receive the message in order to increase its contribution value. The same theory also applies to sender node N.

Thus, when node N chooses to send a message and node M chooses to receive the message, the elements S_3 and S_1 in the resource consumption matrix are both greater than zero, and both nodes N and M can obtain maximum benefit at the same time.

That is to say, when the resource consumption function value is greater than zero, it will increase the value of the contribution function, so that the messages in their buffer will have a greater probability of being received in process of forwarding. So, after two nodes to establish a connection, the sending node will try to send messages and the receiving node will try to choose to receive messages to increase the node's own contribution function value.

TABLE 2. Simulation parameter settings.

Parameter	Parameter settings
Simulation scenarios size	10000 m×10000 m
Simulation time	20 day
The number of nodes	126
Simulation map	Helsinki
Node movement model	SPM
Buffer size	10 M
Message size	100 K-1 M
Message creation interval	20-30 s
TTL	180 min
Node initial energy	500 KmAh
E^{Lis}	1 mAh/s
Es	1.5 mAh/s
Er	2 mAh/s

V. SIMULATION AND RESULTS

In order to validate the ARAG algorithm, this study simulates the proposed algorithm by using the ONE [25] simulator and compares the ARAG algorithm with traditional Epidemic routing algorithm [13], Prophet routing algorithm [14] and GTMEA algorithm [24].

A. SIMULATION ENVIRONMENT

The nodes in this paper will consist of pedestrians, taxis, and trams, the detailed parameters are listed in Table 2. Among them, the energy of taxis and trams is assumed to be infinite, the energy of pedestrian is assumed to be limited, the node's energy consumption can be described by equation (14). Simultaneously, in order to be able to fully simulate the real-time living environment, we establish schools, parks, cafe houses and other public places in the simulation map.

$$E_b = E^{Lis} + Es + Er, \quad (14)$$

where E^{Lis} is the energy consumed by the listening channel. Es is the energy consumed per second when to send

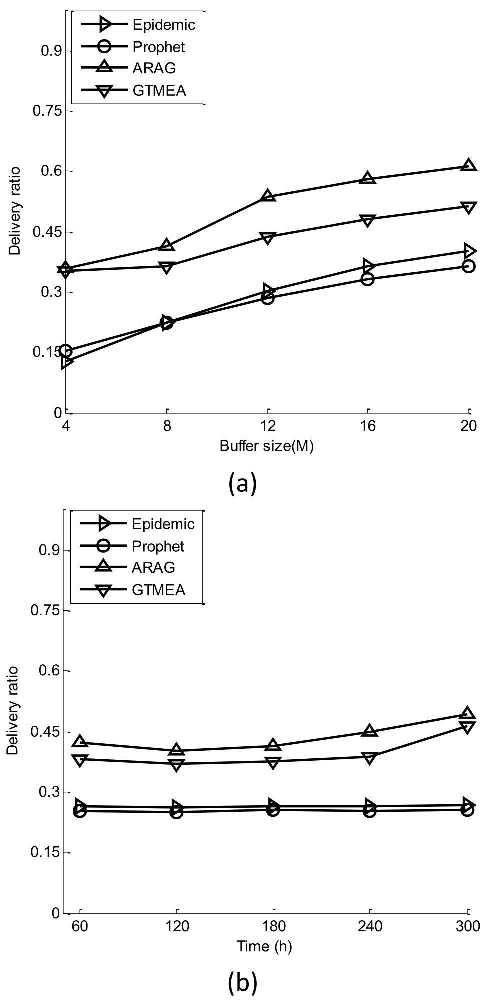

FIGURE 1. Delivery ratio with varying buffer size and simulation time.

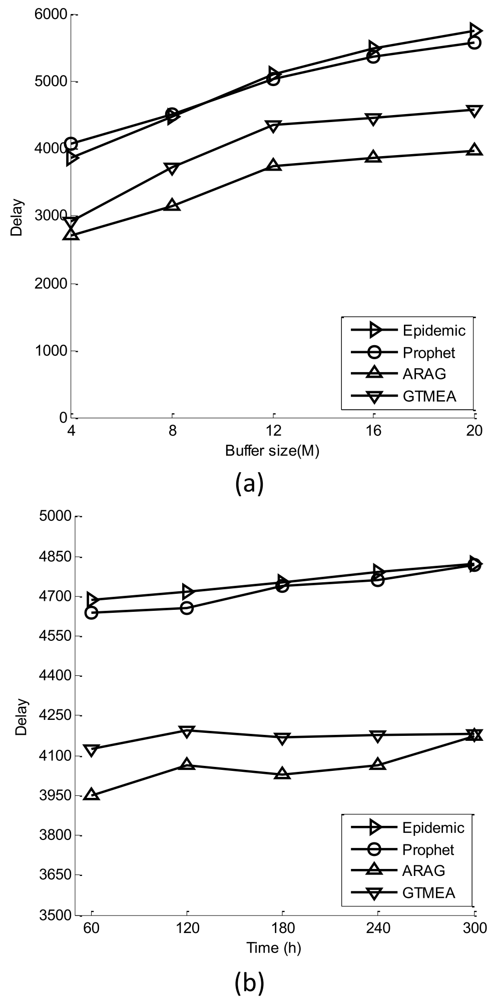

FIGURE 2. Delay with varying buffer size and simulation time.

messages, and Er is the energy consumed per second when to receive messages.

B. PERFORMANCE INDICATORS

In this paper, we will compare the ARAG algorithm with the traditional routing algorithm from three aspects, which are the delivery ratio, average delay, and network overhead. The delivery ratio is the ratio of the number of messages successfully delivered to the destination node and the total number of messages created. The average delay is the ratio of the sum of the time of all delivered messages from the source node to the destination node and the total number of messages. The network overhead is the ratio of the number of messages that are relayed in the network to the number of messages that have been successfully delivered and the number of messages that have been successfully delivered.

C. SIMULATION RESULTS AND ANALYSIS

1) DELIVERY RATIO

Fig. 1a illustrates the changes in the delivery ratio when the buffer size is varied from 4–20 M. As can be seen from Fig. 1a, with the increase in node buffer size from 4M to 20M, the delivery ratios of these algorithms gradually increase. The delivery ratios of the Epidemic and Prophet algorithms are closely matched and varies between 0.1 and 0.4, the delivery ratio of the GTMEA algorithm varies between 0.3 and 0.5, and the delivery ratio of the ARAG algorithm varies between 0.33 and 0.62, which is significantly greater than the three compared algorithms. Fig. 1b depicts the changes in delivery ratio when the simulation time is varied from 60–300 h. It can be seen from Fig. 1b that the delivery ratios of the Epidemic and Prophet algorithms are closely matched, maintain between 0.2 and 0.3, the delivery ratio of the GTMEA algorithm maintains between 0.3 and 0.45, and the delivery ratio of the ARAG algorithm is higher than all of them. Overall, the delivery ratio of the ARAG algorithm is higher than the Epidemic, Prophet and GTMEA algorithms. This is mainly because the message is always forwarded to a relay node with a greater probability of encountering the destination node in the process of selection relay node, and the ARAG algorithm also provides a buffer management strategy that allows messages to be discarded properly, thus increasing the delivery ratio of the messages.

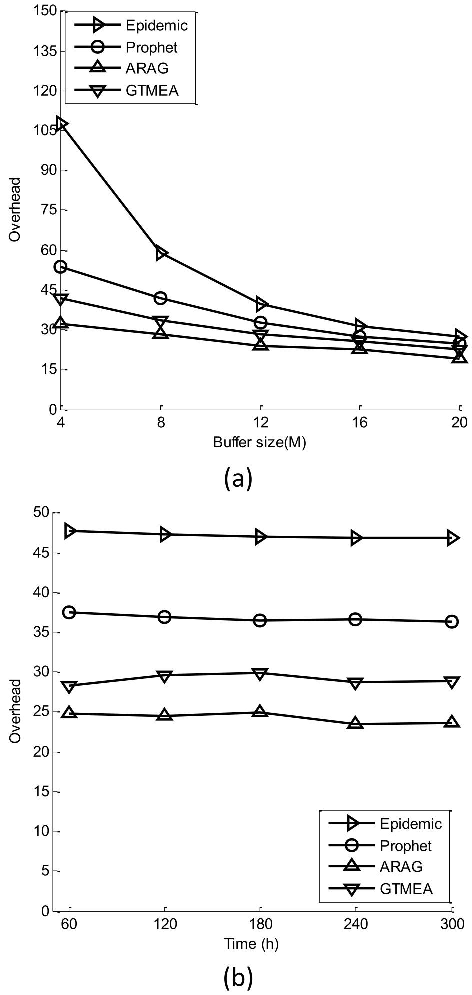

FIGURE 3. Overhead with varying buffer size and simulation time.

2) AVERAGE DELAY

Fig. 2a illustrates the changes in delay when the buffer size is varied from 4–20 M. As can be seen from Fig. 2a, on increasing the node buffer size, the message delays of all the four algorithms increase. The delays of the Epidemic and Prophet algorithms are closely matched, the delay of the GTMEA algorithm is lower than the Epidemic and Prophet algorithms, and ARAG algorithm is the lowest of all. Fig. 2b displays the changes in delay when the simulation time is varied from 60–300 h. As can be seen from Fig. 2b, the delays in the Epidemic and Prophet algorithms are closely matched and relatively stable during the period 60–120 h. The delays in the GTMEA and ARAG algorithms show better performance than Epidemic and Prophet algorithms. The delays of the ARAG algorithm tend decrease when the simulation time is varied from 120–180 h. Overall, the ARAG algorithm shows the best performance in terms of delay. This is mainly because, in the process of forwarding message, by encouraging a node to forward messages to other nodes, the time required for a single message to reach the destination node is reduced, therefore reducing the time that it stays in the network.

3) NETWORK OVERHEAD

Fig. 3a illustrates the changes in the network overhead when the buffer size is varied from 4–20 M; as can be seen from Fig. 3a, when the buffer size increases from 4–20 M, the overheads of the four algorithms are all decreasing. The network overhead of the Epidemic algorithm is the largest; the network overhead of the ARAG algorithm is less than that of the other three compared algorithms. Fig. 3b depicts the changes in the network overhead when the simulation time is varied from 60–300 h. During the simulation period 60–300 h, there are relatively fewer changes in the network overhead of all four algorithms; the overhead of the Epidemic algorithm is the largest, whereas that of the ARAG algorithm is the least. In total, the network overhead of the ARAG algorithm is less than that of the other three compared algorithms. This is mainly because the ARAG algorithm transmits messages to a node with a higher probability of encountering the destination node and by limiting the number of copies of a single message, the total number of message in the network is reduced. Therefore, the network overhead is reduced.

VI. CONCLUSION

This paper presents a routing and management algorithm based on incentive mechanism, which is used to solve the issue of nodes with selfishness. The algorithm aims to optimize the performance of the network by the way of encouraging the sending node to send messages or the receiving node to receive messages to increase the probability that the message it carries is received by a relay node during the process of forwarding message. Meanwhile, the algorithm sets the number of copy thresholds for a message to limit the number of copies of the message, thereby reducing the number of copies of the total message in the network. Simulation results show that the algorithm is superior to the Epidemic algorithm, the Prophet algorithm and the GTMEA algorithm in terms of the delivery ratio, the average delay, and the network overhead.

REFERENCES

[1] K. Fall and S. Farrell, "DTN: An architectural retrospective," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 828–836, Jun. 2008.
[2] A. P. Silva, S. Burleigh, C. M. Hirata, and K. Obraczka, "A survey on congestion control for delay and disruption tolerant networks," *Ad Hoc Netw.*, vol. 25, pp. 480–494, Feb. 2015.
[3] J. Hur and K. Kang, "Secure data retrieval for decentralized disruption-tolerant military networks," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 16–26, Feb. 2014.
[4] S. M. Tornell, C. T. Calafate, J.-C. Cano, and P. Manzoni, "DTN protocols for vehicular networks: An application oriented overview," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 868–887, 2nd Quart., 2015.
[5] G. Hirakawa, N. Uchida, Y. Arai, and Y. Shibata, "Application of DTN to the vehicle sensor platform CoMoSE," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, Mar. 2015, pp. 490–493.
[6] M. Martonosi, "Mobile sensing: Retrospectives and trends," *GetMobile, Mobile Comput. Commun.*, vol. 20, no. 1, pp. 14–19, 2016.
[7] A. Picu and T. Spyropoulos, "DTN-Meteo: Forecasting the performance of DTN protocols under heterogeneous mobility," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 587–602, Apr. 2015.
[8] D. Wu, Y. Wang, H. Wang, B. Yang, C. Wang, and R. Wang, "Dynamic coding control in social intermittent connectivity wireless networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7634–7646, Sep. 2016.

[9] Z.-Y. Li, P.-P. Wu, Y. Song, and J.-L. Bi, "Intermittent data dissemination using node forwarding capability estimation in vehicle delay tolerant networks," in *Proc. IEEE Symp. Service-Oriented Syst. Eng. (SOSE)*, Mar./Apr. 2016, pp. 119–125.

[10] H. Wang, H. Lv, H. Wang, and G. Feng, "DCAR: DTN congestion avoidance routing algorithm based on tokens in an urban environment," *J. Sensors*, vol. 2017, no. 2, pp. 1–9, 2017.

[11] K. Chen, H. Shen, and L. Yan, "Multicent: A multifunctional incentive scheme adaptive to diverse performance objectives for DTN routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1643–1653, Jun. 2015.

[12] H. Wang, G. Feng, H. Wang, H. Lv, and R. Zhou, "RABP: Delay/disruption tolerant network routing and buffer management algorithm based on weight," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 1, 2018, doi: 155014771875787.

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, Feb. 2008.

[14] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *Mobile Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, Jul. 2012.

[15] H. Wang, H. Wang, G. Feng, and H. Lv, "NWBBMP: A novel weight-based buffer management policy for DTN routing protocols," *Peer-to-Peer Netw. Appl.*, no. 2, pp. 1–7, 2017.

[16] R. E. Azouzi, H. B. A. Sidi, J. Rojas-Mora, and A. P. Azad, "Delay tolerant networks in partially overlapped networks: A non-cooperative game approach," in *Proc. Int. Conf. Bio-Inspired Models Netw., Inf., Comput. Syst.*, Avignon, France, Dec. 2009, pp. 195–202.

[17] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Cooperation in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, Mar./Apr. 2003, vol. 2. no. 7, pp. 808–817.

[18] P. Michiardi and R. Molva, "A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile ad hoc networks," in *Proc. Modeling, Optim. Mobile Ad Hoc, Wireless Netw.*, 2014, pp. 3–5.

[19] W. Saad, Z. Han, R. Zheng, A. Hjorungnes, T. Basar, and H. V. Poor, "Coalitional games in partition form for joint spectrum sensing and access in cognitive radio networks," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 2, pp. 195–209, Apr. 2012.

[20] M. Naserian and K. Tepe, "Dynamic probabilistic forwarding in wireless ad hoc networks based on game theory," in *Proc. IEEE 79th Veh. Technol. Conf. (VTC Spring)*, May 2014, pp. 1–5.

[21] M. Gholap and S. Shaikh, "Information sharing in delay tolerant mobile networks with some incentive and fewest transmissions," in *Proc. IEEE Int. Conf. Adv. Comput. Appl.*, Oct. 2017, pp. 197–201.

[22] O. Brun, R. El-Azouzi, B. Prabhu, and T. Seregina, "Modeling rewards and incentive mechanisms for delay tolerant networks," in *Proc. Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw.*, 2017, pp. 233–240.

[23] T. Seregina, O. Brun, R. El-Azouzi, and B. J. Prabhu, "On the design of a reward-based incentive mechanism for delay tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 453–465, Feb. 2017.

[24] Y. Mao and P. Zhu, "A game theoretical model for energy-aware DTN routing in MANETs with nodes' selfishness," *Mobile Netw. Appl.*, vol. 20, no. 5, pp. 593–603, 2015.

[25] A. Keränenm, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proc. Int. Conf. Simulation Tools Techn. Commun., Netw. Syst. Simutools*, 2009, pp. 1–10.

[26] Z. Jiao, R. Tian, B. Zhang, and C. Li, "DTN routing with back-pressure based replica distribution," *J. Commun. Netw.*, vol. 16, no. 4, pp. 378–384, Aug. 2014.

HEZHE WANG received the M.E. degree from Harbin Engineering University in 2016, where he is currently pursuing the Ph.D. degree. His research interests involve delay tolerant networks and opportunistic networks.

HUIQIANG WANG received the M.E. and Ph.D. degrees from Harbin Engineering University (HEU) in 1985 and 2005, respectively. From 2001 to 2002, he was with Queen's University, ON, Canada, as a Senior Visiting Scholar. He is currently involved in teaching and researching as a Professor and a Doctoral Advisor with HEU. He holds 10 Chinese patents. His research interests involve network security, cognitive networks, and autonomic computing.

FANGFANG GUO received the M.E. and Ph.D. degrees from Harbin Engineering University in 2001 and 2006, respectively. His research interests involve cognitive networks, network security, and network architecture.

GUANGSHENG FENG received the M.E. and Ph.D. degrees from HIT in 2005 and 2009, respectively. He is currently involved in teaching and researching as a Lecturer with Harbin Engineering University. His research interests involve cross-layer design, information sensing, and wireless channel access control.

HONGWU LV received the M.S. and Ph.D. degrees from Harbin Engineering University (HEU) in 2009 and 2011, respectively. He is currently involved in teaching and researching as a Lecturer with HEU. His research interests involve cognitive computing and dependability analysis.